REMARKS

This Amendment is in response to the Office Action dated July 30, 2004. In the Office Action, the Examiner objected to Claim 8 because of informalities. Claim 15 and 23 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claims 1-6, 13, 16-22, and 31-37 are rejected under 35 U.S.C. § 102(e) as being anticipated by DeKoning, U.S. Patent No. 6,304,942 (hereinafter *DeKoning*). Claims 7-12, and 39-40 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *DeKoning* in view of Pence, U.S. Patent No. 6,199,146.

Claims 8, 15, 23, and 31 are amended as shown above. Specifically, claim 8 has been amended to correct its informality, while claims 15 and 23 have been amended to remove the word "substantially". Independent claim 31 has been amended to more clearly recite features of the claimed invention. Claims 1-40 remain pending in the application. For the reasons set forth below, the Applicants respectfully request reconsideration and allowance of all pending claims.

CLAIM REJECTIONS - 35 U.S.C. § 102

A claim is anticipated only if each and every element of the claim is found in a single reference. M.P.E.P § 2131 (citing *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628 (Fed. Cir. 1987)). "The identical invention must be shown in as complete detail as is contained in the claim." M.P.E.P. § 2131 (citing *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226 (Fed. Cir. 1989)).

In support of the rejection of independent claims 1 and 22 over *DeKoning*, the Examiner states,

> Regarding Claims 1 and 22, DeKoning discloses a file system with the ability to add additional disk drives to disk arrays ("adding a new file storage device..."). In accomplishing this, the system reconfigures the file system and distributes data across both the new disk and the previous disk ("migrating a portion of the files ...") thus, reconfiguring the file system ("new file system configuration..."). W[h]ile this is happening, the controller continues to present the same logical volumes to the host ("not

to affect file access operations...") See Column 5, lines 40-56. Since the system is a RAID system, the disks are partitioned into data stripes ("Partitioning the storage space...") where each data stripe is of the same size ("each fragment, on average, comprises..."). It is understood that the system in question can be placed in a network where the host system is remote to the Storage Arrays (See Figure 3).

DeKoning discloses a method for providing an upgrade path for an existing data storage system. As stated in the abstract,

> A system and method upgrade an original data storage system into an enhanced data storage system. The original data storage system includes an original storage array controller device and an original storage array. The original data storage system is initially connected to a host system via a host bus and presents to the host system an original logical volume mapping associated with a logical identifier and mapped to a physical data set stored in individual storage devices in the original storage array. An enhanced storage array controller device is operatively connected to the host system via the host bus, and the original storage array controller device and a storage connection device coupled to another storage array are operatively connecting to the enhanced storage array controller device via an intermediate bus. The physical data set of the original storage array is redistributed across a plurality of storage arrays, including the original and the other storage arrays. The enhanced storage array controller device is reconfigured to present to the host system a logical volume mapping associated with the logical identifier mapping to the physical data set redistributed across the plurality of storage arrays.

Claim 1 recites the following elements:

1. A method for incrementally scaling a file system, comprising:

adding a new file storage device to a file system having a storage space comprising at least one other file storage device having a plurality of directories and files stored thereon to form a new file system configuration; and

migrating a portion of the files from said at least one other file storage device to the new file storage device ***while hiding such migration from client applications that access files from the file system so as to not affect file access operations requested and performed by the client applications during the migration***.

(Emphasis added).

The emphasized element in claim 1 is very important to the operations of the scalable network file system of the present invention, and is one of the characteristics that distinguishes the present invention over the prior art. What this language means is that the migration is performed in a manner that is transparent to the client applications – that is, the client applications are unaware the migration is occurring, and their file access operations are unaffected during the migration. Importantly, this means that a given application has access its files (stored on the file system) throughout *all* of the migration process.

This condition does not exist under *DeKoning*. For example, *DeKoning* discloses the operation of one embodiment beginning at Col. 7, line 42, as follows.

> FIG. 4 illustrates a method of upgrading an original data storage system by adding more capacity in an embodiment of the present invention. It should be understood that it is not necessary to add capacity to upgrade an original data storage system. For example, additional channels may be added to an original data storage system to increase availability of existing storage. Alternatively, a higher level of RAID may be achieved by upgrading an original data storage system to entry level bridge RAID controllers to enterprise RAID controllers, for example.
>
> Operation 400 starts the upgrade process. The process starts with an original storage array controller device connected to a host system. An original storage array is coupled through the original storage array controller device to a host system (see FIG. 1). It is understood that the original storage array controller device of FIG. 1 is configurable to support a pass-through mode. The original storage array stores metadata defining the configuration of the logical volume mappings in the storage array. **In operation 402, the original storage array controller device and the original storage array are disconnected from the host system**. Operation 404 connects an enhanced storage array controller device to the host system. Preferably, the enhanced storage array controller device includes at least one of an increased number of disk channels, higher performance capability, or increased RAID functionality. In operation 406, the original storage array controller device (and the original storage array) is connected to the enhanced storage array controller device. Operation 408 connects a storage connection device and an additional storage array to the enhanced storage controller device. The storage connection device may include without limitation a storage array controller device set in pass-through mode, a PCI RAID controller device set in pass-through mode, or an environmental services monitor (ESM). (Emphasis added)

Clearly, this embodiment does not read on "***not affect[ing] file access operations requested and performed by the client applications during the migration.*** The file access is absolutely affected. As soon as the original storage array controller device is disconnected, none of the files stored on devices that were previously connected to that controller device may be accessed. This, in turn, could cause client applications to crash (especially if the original storage array controller device was disconnected in the middle of a file write), or at the very least, would cause some sort of write or read error to be presented to a user of such a client application if a read or write request was made while there was no storage array controller device that was connected to the storage array.

DeKoning also discloses a technique for redistributing data across existing and added storage arrays, beginning at Col. 8, line 41:

> FIG. 5 depicts a flowchart of operations for redistributing data from two existing ELB storage arrays when an additional storage array is added in an embodiment of the present invention. The original data storage system is assumed to be similar to that as shown in FIG. 1. In operation 500, the redistribution process starts, preferably in reaction to a detection of added storage capacity from an additional storage array or to a host command. In operation 502, the enhanced storage array controller device reads the metadata from the original storage arrays to determine the logical volume mappings supported by the storage arrays.
>
> For each drive group, the enhanced data storage array controller device in operation 504 locates the first two storage devices (i.e., A1 and A2) of the first storage array (e.g., storage array 108 in FIG. 1) having overlapping channels with respect to the other storage devices in the drive group. An "overlapping channel" refers to a single channel coupled to multiple drives in the same drive group. **In operation 506, the enhanced storage array controller device copies the data on the storage devices A1 and A2 on different channels to the additional storage array. The two devices A1 and A2 in the first storage array are now available to store new data**.
>
> In operation 508, the enhanced storage array controller device locates the first two storage devices B1 and B2 on the second storage array (e.g., storage array 110 in FIG. 1) having overlapping channels in the storage array with respect to the other storage devices in the drive group**. In operation 510, the enhanced storage array controller copies the data on the storage devices B1 and B2 on different channels to**

> **the additional storage array. The two devices B1 and B2 in the second storage array are now available to store new data**.
>
> In operation 512, the enhanced storage array controller device locates the next two storage devices C1 and C2 on the first storage array having overlapping channels in the storage array with respect to the other storage devices in the drive group. **In operation 514, the enhanced storage array controller copies the data on the storage devices C1 and C2 on different channels to the storage devices B1 and B2. The two devices C1 and C2 in the first storage array are now available to store new data**. (Emphasis Added)

Clearly, this embodiment employs copying files from existing to new storage devices. However, there is absolutely no provision for dealing with accessing a file while it is being copied, thus this embodiment again does not read on "***not affect[ing] file access operations requested and performed by the client applications during the migration.*** The file access is absolutely affected. Suppose a write access is attempted while a file is being copied. One of several results might occur. The write access would be denied by the controller itself or an operating system level device driver for the controller (depending on the operating system architecture). This denial would be propagated to the client application via the operating system. Thus, the client application would not be able to access the file. A similar result would occur with a read access.

Now suppose under the copying scheme a write access was allowed. For example, an image of the original file was copied to a buffer, freeing the original file, and then the buffer was copied to the destination file on the new storage device. The write access would change the contents of the original file. Meanwhile, the original file would be copied to the destination file on the new storage device. This result would produce two different files, which is unacceptable.

With respect to the Examiner's assertion that the text disclosed in Column 5, lines 40-56 reads on the claim element "not to affect file access operations...", applicants respectfully assert this is clearly not the case. Col. 5, lines 40-56 reads,

> A user in such a situation has several alternatives. First, for example, a user may add additional disk drives to one or more of the disk arrays in order to increase storage capacity. Furthermore, employing a feature called Dynamic Capacity Expansion (DCE), the user can automatically reconfigure the data on the original disks of a storage array, distributing it across both the original storage disks and the added storage disks. After reconfiguration, all data is distributed evenly across all storage disks in the storage array, while the RAID controller continues to present the same logical volumes to the host system. Likewise, all unused capacity appears evenly distributed across all drives following the last LUN. Therefore, unused capacity may be used to create additional LUNs. Generally, DCE is limited by the maximum number of storage disks a particular RAID controller can support and by the number of channels available to access the plurality of storage disks in the array. (Emphasis added)

The foregoing text merely discusses a technique for increasing storage capacity that was available in the prior art at the time *DeKoning* was filed (i.e., outside of what is disclosed by *DeKoning's* invention). DCE is a feature that is built-into various RAID controllers, such as RAID controller made by Adaptec, for example. (It is noted that LSI logic, the assignee of *DeKoning* patent, is a chip supplier to Adaptec, as well as other RAID controller manufacturers.) The emphasized portion of the text identifies a limitation with this conventional approach that is identified by *DeKoning* – that being the expansion is limited to the capabilities of the existing RAID controller.

The applicants agree that after DCE operations have been completed, the file access is the same (from the applications viewpoint) as before the expansion. However, DCE provides no measures for migrating files in a manner that is transparent to the client applications *during the migration*.

DeKoning clearly does not disclose any mechanism for dealing with file accesses during migration. In contrast, embodiments of the present invention provide a novel file-locking scheme that immediately releases locks (the locks are "stolen back" by clients) on files that are requested to be accessed in a manner that appears transparent to the client applications. The operating system does not receive an indication from the file system that the file is not accessible, and thus such information is not propagated back to the client application by the operating system. Furthermore, embodiments of the present invention can determine if a file access attempt occurs during a migration, such that a modified file will be migrated after it obtains a new lock (in response to it no longer being accessed by a client application application). Details of this process are generally discussed with reference to the flowchart of Figure 9.

DeKoning clearly does not teach or fairly suggest each and every element of independent claim 1. Accordingly, the rejection of claim 1 is improper, and should be withdrawn. Furthermore, the rejection of each of claims 2-21, which depend either directly or indirectly from independent claim 1, is improper and should be withdrawn for at least the same reasons presented above in support of the patentability of claim 1.

Independent claim 22 recites a similar element to that in claim 1 with respect to the transparency of file access during migration. In particular, claim 22 recites,

"migrating files among said plurality of files from network storage devices on which they are initially stored to other network storage devices corresponding to the fragment they are assigned to ***in a manner such that the migration of files are undetectable to client applications that access the network file system***." (Emphasis added)

Clearly, *DeKoning* does not disclose a migration process that meets this claim limitation, as discussed above. Under *DeKoning*, the client applications would absolutely detect if a file was not accessible during the migration due to the file currently being copied. Accordingly, the rejection of independent claim 22 is improper, and should be withdrawn. Furthermore, the rejection of each of claims 23-30, which depend either directly or indirectly from independent claim 22, is improper and should be withdrawn for at least the same reasons as claim 22.

With respect to the rejection of independent claim 31, the Examiner states,

> Regarding Claim 31, DeKoning discloses a file system, comprising a host system 102 ("client machine ...") with the ability to add additional disk drives to disk arrays ("adding a new file storage device..."). In

accomplishing this, the system reconfigures the files system and distributes data across both the new disk and the previous disk ("migrating a portion of the files ...") thus, reconfiguring the file system ("new file system configuration...'). W[h]ile this is happening, the controller continues to present the same logical volumes to the host ("not to affect file access operations..."). See Column 5, lines 40-56. Since the system is a RAID system, the disks are partitioned into data stripes ("partitioning the storage space..."). It is understood that the system in question can be placed in a network where the host system is remote to the Storage Arrays (See Figure 3). The storage devices in the storage arrays are accessed via enterprise storage controller 310 (file system protocol) and the abstraction layer is provided made of storage controller 310 and ELB RAID Controller 306, which presents the files system as a virtual file system. Presentation as a virtual fie system takes place when the RID controller continues to present the same number of logical volumes to the host even when additional volumes have been added (column 40-56). The abstraction layer is informed of new file system configuration when the enterprise storage controllers read metadata to reconstruct the logical mapping ("mapping data...") of data and volumes (Column 6, line 57 – Column 7, line 17).

Amended claim 31 now recites:

31. A network file system comprising:

a plurality of file storage devices connected to a network;

at least one client machine having a processor, connected to the network in communication with said plurality of file storage devices, operating under an operating system running on the processor and hosting a client application that runs on the operating system and accesses files stored on the network file storage system;

a storage abstraction layer comprising <u>at least one module running on the processor</u> and providing an interface to the client application that virtualizes the network file system such that it appears as a virtual storage space comprising a set of virtual directories and files to the client application; and

mapping data, accessible to the storage abstraction layer, that maps the virtual directories and files to corresponding physical directories and files stored on said plurality of file storage devices. (Emphasis added)

DeKoning does not disclose a "*a storage abstraction layer comprising at least one module running on the processor …*" The asserted equivalent to the storage abstaction layer is storage controller 310 and ELB RAID Controller 306. These are hardware devices. Clearly, they are not software modules that are run on processor. Therefore, *DeKoning* does not disclose, teach, or suggest all of the elements of amended independent claim 31. Accordingly, independent claim 31 is patentable over the cited art.

* * * * *

Overall, none of the references singly or in any motivated combination disclose, teach, or suggest what is recited in the independent claims. Thus, given the above amendments and accompanying remarks, independent claims 1, 22, and 31 are now in condition for allowance. The dependent claims that depend directly or indirectly on these independent claims are likewise allowable based on at least the same reasons and based on the recitations contained in each dependent claim.

If the undersigned attorney has overlooked a teaching in any of the cited references that is relevant to the allowability of the claims, the Examiner is requested to specifically point out where such teaching may be found. Further, if there are any informalities or questions that can be addressed via telephone, the Examiner is encouraged to contact the undersigned attorney at (206) 292-8600.

Charge Deposit Account

Please charge our Deposit Account No. 02-2666 for any additional fee(s) that may be due in this matter, and please credit the same deposit account for any overpayment.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Date: Sept 9, 2004

R. Alan Burnett
Reg. No. 46,149

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1030

Attachments: Substitute Drawing Sheet for Figure 9.